УДК 519.7
Юхта О.А., Кізим С.О., Ройко О.О., Ройко О.Ю.
Волинський коледж Національного університету харчових технологій, м.Луцьк, Україна

АНАЛІЗ ОНЛАЙН-АЛГОРИТМІВ НА ПРИКЛАДІ ЗАДАЧІ ПРО ПРОКАТ ЛИЖ З ДВОМА ОПЦІЯМИ

О.А. Юхта, С.О. Кізим. О.О. Ройко, О.Ю. Ройко. Аналіз онлайн-алгоритмів на прикладі задачі про прокат лиж з двома опціями. В статті розглядаються онлайн-алгоритми, що обробляють дані згідно міри їх надходження, зокрема, на прикладі задачі про прокат лиж. Подано аналіз класичної задачі прокату лиж та її узагальнення, коли відсутня опція «чистої купівлі», а також описані робота оптимального алгоритму для задачі та обгрунтування його оптимальності.
Ключові слова: онлайн-алгоритм, дискретна оптимізація, дискретне програмування, задача про прокат, конкурентний аналіз.
Рис. 3. Літ. 5.

А.А. Юхта, С.А. Кизым. О.А. Ройко, А.Ю. Ройко. Анализ онлайн-алгоритмов на примере задачи о прокате лыж с двумя опциями. В статье рассматриваются онлайн-алгоритмы, обрабатывающие данные по мере их получения, в частности, на примере задачи о прокате лыж. Подан анализ классической задачи проката лыж и её обобщение, когда отсутствует опция «чистой покупки», а также описаны работа оптимального алгоритма для задачи и обоснование его оптимальности.
Ключевые слова: онлайн-алгоритм, дискретная оптимизация, дискретное программирование, задача о прокате лыж, конкурентный анализ.

O.A. Yukhta, S.O. Kizym. O.O. Roiko, O.Yu. Roiko. Online algorithms analysis on the example of the ski rental problem with two options. The article discusses online algorithms that process its input piece-by-piece in a serial fashion, in particular, on the example of the ski rental problem. The analysis of the classical problem of ski rental and its generalization, when there is no "pure purchase" option, is given and the work of the optimal algorithm for the problem and the justification for its optimality are described.
Keywords: online algorithm, discrete optimization, discrete programming, ski rental, competitive analysis.

Системна програма, що контролює роботу деякої операційної системи, повинна генерувати певні дії (приймати рішення, записувати певні дані на вихід) до того, як надійдуть всі дані. Алгоритми, що виконують такі дії, називаються алгоритмами, що працюють в режимі реального часу, або онлайн-алгоритмами. Як правило, онлайн-алгоритм отримує послідовність запитів. По кожному із запитів він повинен надати деякий сервіс до того, як отримає наступний запит.

Задача прокату лиж є одним з основних питань онлайн-обчислень. В оригінальному формулюванні нам потрібно використовувати ресурс протягом невідомого проміжку часу. Є два варіанти зробити це: або заплатити за одиницю ресурсу та мати можливість використовувати його будь-яку кількість часу (опція купівлі), або оплачувати ресурс пропорційно часу використання (опція прокату). Якщо час використання менший за умовну часову одиницю, то оптимальним є варіант прокату, а в іншому випадку оптимальним варіантом являється купівля у «нульовий» момент часу. Очевидно, що кращою детермінованою онлайн-стратегією є використання опції прокату за одиницю часу, а потім купівля ресурсу. Тобто стратегія ніколи не платить оптимальне значення більше, ніж двічі. У гіршому випадку гра зупиняється відразу після того, як стратегія виконає купівлю ресурсу. Також відомо, що найкраща рандомізована онлайн-стратегія забезпечує очікуване співвідношення оптимальності $\frac{e}{e-1} \approx 1{,}58$. Стратегія полягає у випадковому виборі часу $t \in [0;1]$ у відповідності до функції щільності ймовірності $f(t) \in \frac{e^t}{e-1}$ та переході з опції прокату на опцію купівлі в момент часу *t*, якщо гра ще не закінчилась. У даному випадку розглядається деяке узагальнення вихідної задачі, коли немає варіанту «чистої купівлі», тобто може бути виконана деяка оренда. Очевидно, що реальне життя несе у собі багато прикладів, що відносяться саме до цієї категорії.

Є два варіанти: перший, так званий «схил 1», полягає тому, щоб заплатити одиницю вартості за одиницю часу. В іншому випадку, так званому «схил 2», оплата виконується в одиницях вартості за кожну одиницю використання часу для деякого реального $0 \leq a < 1$. Гра починається, коли користувач знаходиться на схилі 1, і він може перейти на схил 2 у будь-який момент часу по собівартості $1 - a$, але подальші зміни не допускаються. В деякий момент часу t гра зупиняється. Загальна вартість для користувача у момент часу t – це час, затрачений на схил 1, додати $1 - a$, якщо був здійснений перехід на схил 2, та додати час, затрачений на схил 2. Задача алгоритму – визначити момент, коли потрібно перейти на схил 2. Розглядаються рандомізовані алгоритми, визначається вартість для користувача в момент часу t, як очікувана вартість алгоритму, коли час зупинки становить t. Відзначимо, що оптимальною вартістю є t, якщо гра зупиняється в момент $t \leq 1$ або $(1 - a) + at$, якщо $t > 1$. При $0 < a < 1$ вищезазначене формулювання фіксує будь-який випадок, коли вартість для схилу 1 оцінюється як $f(x) = a_1x$, а вартість для схилу 2 рівна $g(x) = a_2x + b$, припускаючи нетривіальність, тобто $0 \leq a_2 < a_1$. При $a = 0$ – це випадок основної задачі прокату лиж. Виявляється, це незначне узагальнення якісно відрізняється від класичного випадку. Наприклад, у класичному прокаті лиж будь-який алгоритм повинен у певний момент перемикнутися на опцію купівлі, у протилежному випадку його вартість зростає без обмежень, а оптимальна вартість залишається сталою. Коли «чиста купівля» недоступна, найкращий онлайн-алгоритм ніколи не зможе перемикнутись на другу опцію. [1, 3]

Узагальнена проблема з орендою лиж може бути зображена на графіку, як дві лінії, що перетинаються на площині, кожна з яких відповідає своїй опції, як показано на малюнку 1. Вісь x відповідає часу, а вісь y відповідає вартості. Варіант оренди відповідає крутій лінії, яка перетинає вісь y ближче до початку координат. Нормалізована модель припускає, що схил оренди перетинає початок координат, і що схили перетинаються в точці (1,1).

Об'єктом подальшого розгляду є наступна теорема.

Теорема 1.1. Розглянемо задачу про прокат лиж, де в опції 1 вартість покупки дорівнює 0 і орендна ставка становить 1 для кожної одиниці часу, а у опції 2 вартість покупки становить $1 - a$ і орендна ставка дорівнює a для кожної одиниці часу. Тоді очікуваний коефіцієнт конкурентоспроможності у цьому випадку є точно $\frac{e}{e-1+a}$.

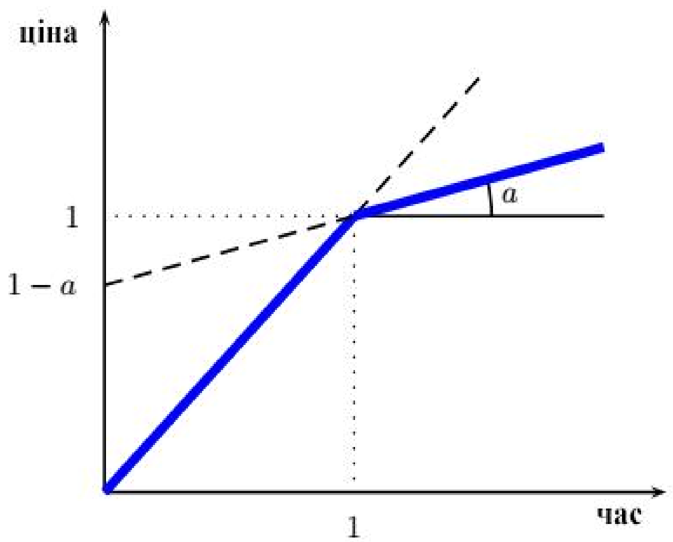

Рис. 1. **Ілюстрація узагальненої задачі про прокат лиж з двома опціями**

Як згадувалося вище, класична задача про прокат лиж, де вартість покупки «схил 1» та вартість прокату «схил 2» дорівнюють нулю, були представлені в [3] при досягненні конкурентних факторів 2 (детерміністичний) та *e/e-1* (рандомізований). Класичну задачу про прокат лиж інколи називають задачею лізингу.

Розглянемо рандомізований онлайн-алгоритм задачі. Нехай $p_1(t)$ і відповідно $p_2(t)$ позначають ймовірність того, що алгоритм використовує схил 1 (або відповідно схил 2) у деякий момент часу t. Потрібно звернути увагу, що $p_2(t) = 1 - p_1(t)$, оскільки алгоритм повинен виконувати «схил 1» або «схил 2» у будь-який момент часу t. Таким чином, алгоритм повністю визначається через $p_1(t)$. У розглянутому нижче алгоритмі спочатку потрібно знайти $p_1(t)$ у термінах *c*, де *c* – конкурентний фактор алгоритму, який ще повинен бути визначений.

Розглянемо час *t*, $0 \le t \le 1$. Очікувана швидкість, з якою алгоритм буде витрачати кошти (тобто, похідна від вартості), буде мати вигляд:

$$1 * p_1(t) + p_2'(t) * (1 - a) + a * p_2(t)$$

Перший терм – це витрата, обумовлена перебуванням на схилі 1. Другий терм – це витрата, пов’язана з інвестиціями у купівлю на схилі 2. Третій терм відповідає за перебування на схилі 2. Потрібно нагадати, що якщо гра зупиняється за час $t < 1$, то оптимальна стратегія знаходиться на схилі 1, а отже показник витрат становить 1. З цього випливає, що алгоритм не може проводитися зі швидкістю, більшою за $c * 1 = c$. Оскільки на даний момент нас цікавить лише найгірший показник конкурентоспроможності, а також оскільки більші витрати дозволяють рухатися швидше від схилу 1 до схилу 2, будемо дозволяти нашому алгоритму витрачати кошти з максимально можливою швидкістю, а саме зі швидкістю, точною *c*. Отже, маємо:

$$\begin{aligned} c &= p_1(t) + p_2'(t) * (1 - a) + a * p_2(t) \\ &= p_1(t) + p_1'(t) * (1 - a) + a * (1 - p_1(t)) \\ &= a + (1 - a) * (p_1(t) - p'(t)) \end{aligned}$$

оскільки $p_2(t) = 1 - p_1(t)$, і отже $p_2'(t) = -p_1'$.

Розглянемо наступні приклади з двома схилами: графіки показують $p_1(t)$ (суцільні лінії) та $p_2(t)$ (пунктирні лінії). У класичному випадку другий схил купується з ймовірністю 1 за час 1. При $a = 0.5$ існує ймовірність $\frac{1}{2e-1} \approx 0.23$ того, що перехід на схил 2 ніколи не буде здійснено.

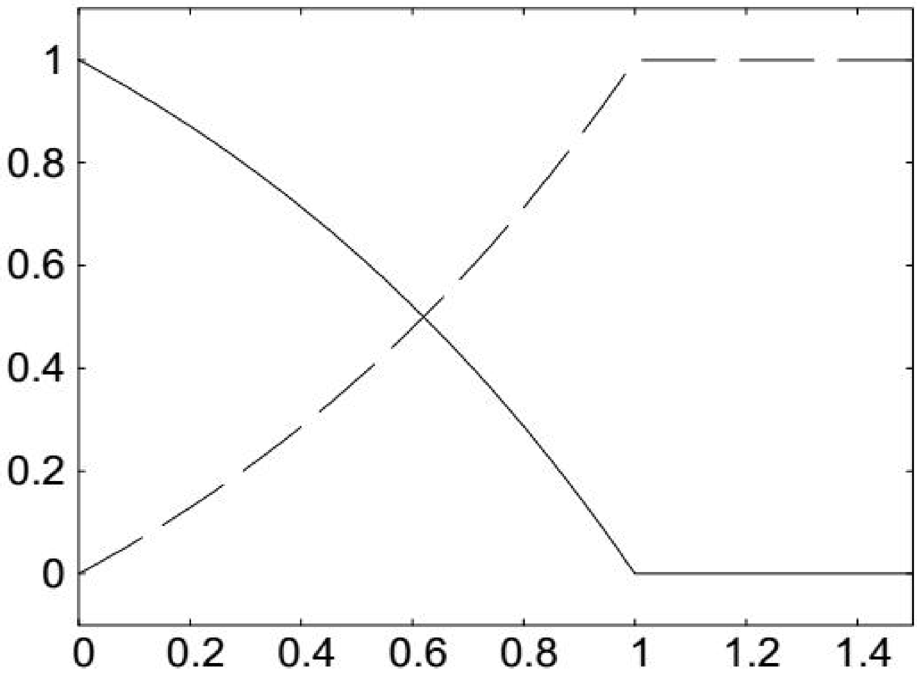

Рис. 2. **Приклад схилів класичного випадку при $a = 0$**

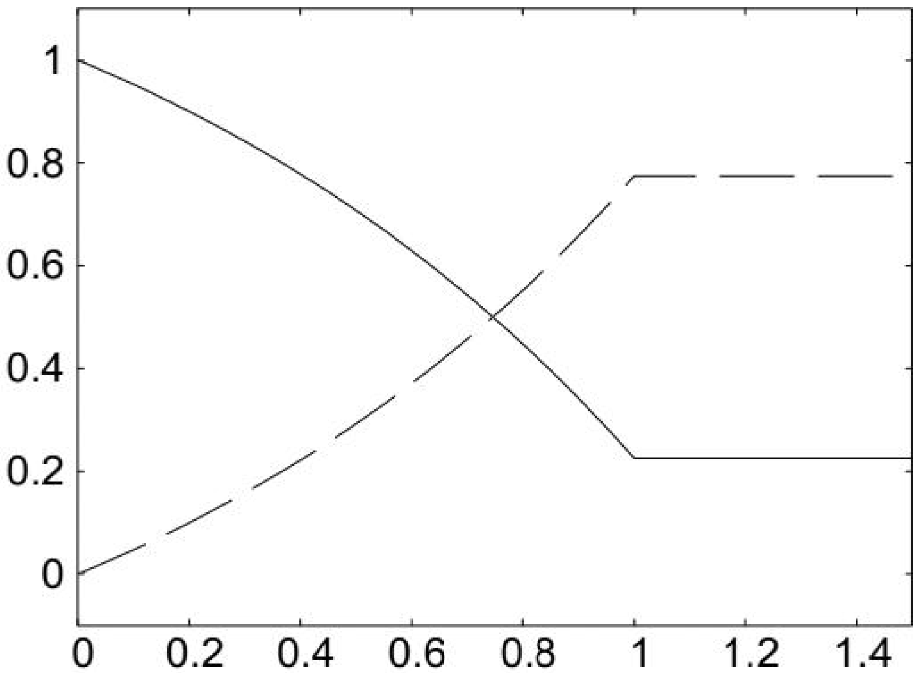

Рис. 3. **Приклад схилів при $a = 0.5$**

За умови, що $p_1(0) = 1$ отримаємо:

$$p_1(t) = \frac{(1-c)e^t + (c-a)}{1-a} \quad (1) \tag{1}$$

Зокрема,

$p_1(1) = \frac{(e-a)-c(e-1)}{1-a}$.

Щоб знайти значення конкурентного співвідношення *c*, розглянемо $t > 1$, де оптимальна стратегія витрачає гроші за курсом *a*. Припускаємо, що алгоритм припиняє купівлю у час 1, тобто якщо перемикання на схил 2 не відбулося за час, рівний 1, то жодне перемикання ніколи не відбудеться. Іншими словами, ми припускаємо, що $p_1'(t) = 0$ при $t > 1$ (доцільність припущення буде підтверджена далі). Беручи його до уваги, будемо мати, що

$$ca = a + (1-a)p_1(t). \tag{2}$$

ставивши $p_1(1)$ з рівняння (1) у рівняння (2), ми можемо знайти рішення відносно *c*, отримуючи, що коефіцієнт конкурентоспроможності алгоритму є

$$c = \frac{e}{e-(1-a)},$$

що доводить верхню оцінку Теореми 1.1. Підставивши значення *c* назад до рівняння 1, ми робимо висновок, що алгоритм характеризується

$$p_1(t) = \begin{cases} \dfrac{a+e-e^t}{e-(1-a)} \text{ при } 0 \le t \le 1 \\ \dfrac{a}{e-(1-a)} \text{ при } t > 1. \end{cases}$$

Альтернативно, функція щільності ймовірності купівлі на схилі 2 є для

та 0 у будь-якому іншому випадку.

Приклади. На рисунках 2 та 3 продемонстровано різницю між класичною задачею про прокат лиж та варіантом, коли відсутня можливість «чистої купівлі». Вони містять оптимальні рандомізовані онлайн-алгоритми для двох випадків: перший – з параметром *a* = 0 (класичний прокат лиж) та інший – з *a* = 0.5. Потрібно відзначити, що у попередньому $p_2(1) = 1$, в той час як у останньому $p_2(1) < 1$. Конкурентне співвідношення для другого прикладу виглядає як $\frac{e}{e-0.5} \approx 1.225$. [1,2,5]

Щоб показати, що описаний алгоритм оптимальний, потрібно побудувати розподіл по входам і продемонструвати, що очікувана вартість будь-якого детерміністичного алгоритму є щонайменше *c* разів оптимальною, де $c = \frac{e}{e-(1-a)}$ є коефіцієнтом конкурентоспроможності нашого алгоритму. Розподіл не залежить від *a*. Зокрема, функція щільності ймовірності зупинки у момент часу *x* становить e^{-x} для $0 \le x \le 1$ та існує ймовірність зупинки 1/*e* у момент часу 2. Тому оптимальна очікувана вартість

$$c_0 \stackrel{\text{def}}{=} \int_0^1 xe^{-x}dx + \left((1-a)+2a\right) * \frac{1}{e} = \frac{e-(1-a)}{e}.$$

Перший член – це вартість, коли гра зупиняється за час 1, другий член – для випадку, коли гра зупиняється у момент часу 2.

Зараз демонструється той факт, що будь-який детерміністичний онлайн-алгоритм має очікувану вартість принаймні 1 для вищевказаного розподілу вхідних даних. Це, згідно з лемою Яо, дозволяє зробити висновок, що жоден рандомізований алгоритм не має конкурентного співвідношення, кращого за $\frac{1}{c_0} = \frac{e}{e-(1-a)}$.

Розглянемо спершу детерміністичний алгоритм, що рухається від схилу 1 до схилу 2 за час $t \in [0,1]$. Тоді очікувана вартість становить

$$\int_0^t xe^{-x}dx + \int_t^1 \left(t+(1-a)+a(x-t)\right)e^{-x}dx + \left(t+(1-a)+a(2-t)\right) * \frac{1}{e}$$
$$= \left(1-\frac{1+t}{e^t}\right) + \left(\frac{1+t}{e^t} - \frac{1+a+t-at}{e}\right) + \frac{1+a+t-at}{e},$$

що ідентично виявляється 1. Перший терм – для випадку, коли алгоритм зупиняється за час *t*, другий терм – для випадку, коли алгоритм зупиняється між часом *t* та часом 1, останній, третій терм – для випадку, коли алгоритм зупиняється за час 2.

Розглянемо наступним алгоритм, що рухається зі схилу 1 на схил 2 за час $1 < t \le 2$. Тоді його очікувана вартість становить

$$\int_0^1 xe^{-x}dx + \left(t+(1-a)+a(2-t)\right) * \frac{1}{e} = \frac{e-2}{e} + \frac{(1+t)(1-a)+2a}{e} > 1$$

для будь-якого $t > 1$ і $a < 1$.

Нарешті, для алгоритму, який переходить на схил 2 за час $t > 2$ будемо мати

$$\int_0^1 xe^{-x}dx + 2 * \frac{1}{e} = \frac{e-2}{e} + \frac{2}{e} = 1,$$

а отже нижня оцінка Теореми 1.1 також доведена. [4,5]

1. Albers S. Online algorithms: a survey / Susanne Albers. // Mathematical Programming. – 2003. – №97. – С. 3–26.
2. Borodin A. Online Computation and Competitive Analysis / A. Borodin, R. El-Yaniv., 1998. – (Cambridge University Press).
3. Competitive snoopy caching / A. R.Karlin, M. S. Manasse, L. Rudolph, D. D. Sleator. // Algorithmica. – 1988. – №3(1). – С. 77–119.
4. Lotker Z. Randomized Algorithms for Multislope Ski Rental / Z. Lotker, B. Patt-Shamir, D. Rawitz. // STACS 2008. – 2008. – С. 503–514.
5. Lotker Z. Ski rental with two general options / Z. Lotker, B. Patt-Shamir, D. Rawitz. // Information Processing Letters. – 2008. – Volume 108, Issue 6 – С. 365–368.